Auto-generation of Passive Scalable Macromodels for Microwave Components using Scattered Sequential Sampling

Krishnan Chemmangat[1], Tom Dhaene[1], Luc Knockaert[1]

[1] K. Chemmangat, T. Dhaene and L. Knockaert are with Ghent University - iMinds, Gaston Crommenlaan 8 Bus 201, B-9050 Gent, Belgium (email: krishnan.cmc@intec.ugent.be, tom.dhaene@ugent.be, luc.knockaert@ugent.be

This paper presents a method for automatic construction of stable and passive scalable macromodels for parameterized frequency responses. The method requires very little prior knowledge to build the scalable macromodels thereby considerably reducing the burden on the designers. The proposed method uses an efficient scattered sequential sampling strategy with as few expensive simulations as possible to generate accurate macromodels for the system using state-of-the-art scalable macromodeling methods. The scalable macromodels can be used as a replacement model for the actual simulator in overall design processes. Pertinent numerical results validate the proposed sequential sampling strategy.

Corresponding author: Krishnan Chemmangat; email: krishnan.cmc@intec.ugent.be; phone: +32 9 331 48 91

I Introduction

Efficient design of electromagnetic (EM) systems using accurate scalable macromodels is an active field of research [1–6]. These scalable macromodels are computationally cheap and they act as a replacement model for the expensive EM solvers thereby reducing the overall computational burden. The scalable macromodeling schemes are also able to preserve system properties such as stability and passivity and hence can be used in time-domain simulations [1–6]. However, these state-of-the-art macromodeling schemes suffer from the fact that the sample distribution over the design variable space should be known a priori based on rules of thumb [1–6].
Several sequential sampling algorithms have been suggested in the literature for automatically building scalable macromodels for the EM systems [7–12]. All of these sampling schemes are global but often fail to guarantee stability and passivity. Preserving system properties is very important especially if the model thus generated is employed in time-domain simulations [13]. Also, for relatively high dimensions the memory requirement for these methods can be relatively high since big matrices has to be solved, limiting their applicability [7–9, 12]. Recently a local tree-based sequential sampling has been proposed in [14] which uses interpolation-based local scalable macromodeling method to build accurate parameterized macromodels. The method is able to preserve

system properties and can also build multi-fidelity models. This means that the designer can already begin the design process once sufficient accuracy is reached for the intermediate model while the model is still being refined. Also the method is implemented as a tree with independent branches for different regions of the design space making it easily expandable and portable to parallel computing platforms.
In this paper, we improve the method of [14], which we refer throughout this paper as *grid method*, on the following aspects:

1 The grid method deals with hyperrectangular regions of the design space which are called *subspaces* in this paper. In the previous method, after performing an edge-based division the algorithm finally divides along the center of the subspace. This is an exploratory step which tends to generate a considerable number of samples per division. This becomes more critical with higher dimension. In this work, the final refinement after the edge-based division is performed using a scattered division with well-conditioned simplicial partitions called path-simplexes [15, 16] reducing the overall complexity of the problem.

2 The grid method requires validation samples to access the accuracy of the terminal subspaces which need not be used further in the final model. In this paper, the validation samples are altogether eliminated by using a level-based check wherein two subsequent levels of models are compared for convergence. This results in a considerable reduction in the overall number of points required.

The refinement on the simplexes can be done in many ways such as dividing along in-center. However, this might lead to the creation of ill-conditioned simplexes called *slivers*. Generation of slivers can be avoided by refining either locally [15, 16] or globally [17, 18]. The local refinement scheme [15, 16] starts from the corner points of an N-cube and then refines it into smaller simplexes in a tree-based way like the sequential sampling method of [14], whereas the global refinement schemes [17, 18] work on a primary Delaunay tessellation and then refine it to improve the condition of simplexes. Hence the local path-simplex method [15, 16] assures a good condition number from the beginning of the sampling process and is suitable for the application of different passivity-preserving scalable macromodeling algorithms on scattered grids [2, 5]. On the other hand, if the global refinement schemes [17, 18] were used, the existing mesh has to undergo global refinement indicating that the local interpolated models may change significantly with a consequent computational burden. So, a path-simplex based refinement is employed in this paper.
However, since the path-simplex method gives more importance to conditioning of the simplexes, it is more a space-filling strategy. So, in this work a hybrid scheme which combines all the benefits of the grid-based refinement and the path-simplex refinement is used to get an efficient sequential sampling strategy requiring less computational resources.
This paper is organized as follows: Section II briefly describes a scalable macromodeling method which is employed in this paper and the grid method of [14]. Section III defines a path-simplex and its well-conditioned refinement with its relative merits and demerits with respect to the grid-based scheme of [14] and states why a hybrid strategy which combines an edge-based and a scattered refinement is required. Section IV demonstrates

how the error can be estimated without having to use expensive validation points. The complete flowchart of the proposed sequential sampling is given with description in Section V followed by three numerical examples in Section VI. Section VII concludes the paper.

II Preliminaries

This section briefly explains a robust scalable macromodeling method used in the paper and also recapitulate the grid-based sequential sampling method of [14].

A) Passivity Preserving Scalable Macromodeling

In this work, we use one of the local scalable macromodeling schemes which use the Vector Fitting (VF) technique [19] to build frequency-dependent rational models called *root macromodels* at the selected design space samples and then parameterize them, see [1–6]. These methods preserve stability and passivity over the complete design space, and therefore are suitable for time-domain simulations. The scalable macromodeling process starts with a set of multivariate data samples $\{(s,\vec{g})_k, \mathbf{H}(s,\vec{g})_k\}_{k=1}^{K_{tot}}$ which depends on frequency and additional design variables. From these data samples, a set of *root macromodels* in pole-residue form are built for a set of design space samples $\vec{g}_k$ by means of VF yielding a set of *root macromodels* $\mathbf{R}(s,\vec{g}_k)$. Stability and passivity are enforced using robust standard techniques [19, 20], resulting in a set of stable and passive *root macromodels*. The next step of these scalable macromodeling algorithms is the parameterization of the set of *root macromodels* $\mathbf{R}(s,\vec{g}_k)$.
In [1, 2, 5], a scalable macromodel is built by interpolating a set of *root macromodels* at an input-output level, while in [3, 4], both poles and residues are parameterized by interpolating the internal state-space matrices, resulting in higher modeling capability with respect to [1, 2]. In [6], a novel enhanced interpolation of *root macromodels* at an input-output level is described, which is based on the use of some coefficients: one coefficient as a multiplicative factor at the input/output level of the system and the other coefficient as a compression or expansion term for the Laplace variable s. It results in high modeling capability and robustness and it is used in our sequential sampling.
To understand the macromodeling method of [6], let us consider a 2 variable design space region Ω_l, $l = 1,\ldots,L$ given in Fig. 1. A two variable description is presented here for clarity and ease of notation, even though the method is general for any dimension N of the design space. The rational *root macromodels* $\mathbf{R}^{\Omega_l}(s,\vec{g}_i{}^{\Omega_l})$, $i = 1,\ldots,2^N$ contained in the N-box region Ω_l are represented in a pole-residue form:

$$\mathbf{R}^{\Omega_l}(s,\vec{g}_i{}^{\Omega_l}) = \sum_{p=1}^{P_i^{\Omega_l}} \frac{C_{p,i}^{\Omega_l}}{s - a_{p,i}^{\Omega_l}} + D_i^{\Omega_l} \;\; ; p = 1,\ldots,P_i^{\Omega_l} \tag{1}$$

where $C_{p,i}^{\Omega_l}$ represents the residue matrices, $a_{p,i}^{\Omega_l}$ the pole $P_i^{\Omega_l}$. $D_i^{\Omega_l}$ is the direct-term matrix.
Later, the parametric macromodeling is applied on these N-box regions Ω_l. The design space region Ω_l is defined by four bounding corners $\vec{g}_1{}^{\Omega_l} = (g_1^1, g_2^1)$, $\vec{g}_2{}^{\Omega_l} = (g_1^2, g_2^1)$,

$\vec{g}_3^{\,\Omega_l} = (g_1^1, g_2^2)$, and $\vec{g}_4^{\,\Omega_l} = (g_1^2, g_2^2)$ as in Fig. 1. Each corner possesses a different *root macromodel* $\mathbf{R}^{\Omega_l}(s, \vec{g}_i)$, $i = 1, \ldots, 4$. We will discuss the interpolation of the *root macromodels* next. For simplicity and ease of notation we omit the superscript Ω_l. In [6],

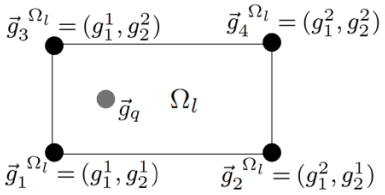

Figure 1: A two dimensional design space with four *root macromodels*.

one amplitude scaling and one frequency scaling coefficient (α_1, α_2) are calculated using the optimization

$$(\alpha_{1,ij}^*, \alpha_{2,ij}^*) = \underset{(\alpha_{1,ij}, \alpha_{2,ij})}{\operatorname{argmin}} \left[\operatorname{Err}(\tilde{\mathbf{R}}^j(s, \vec{g}_i), \mathbf{R}(s, \vec{g}_j)) \right] . \; i = 1, \ldots, 4; \; j = 1, \ldots, 4; \tag{2}$$

In (2), $\tilde{\mathbf{R}}^j(s, \vec{g}_i) = \alpha_{1,ij} \mathbf{R}(s\alpha_{2,ij}, \vec{g}_i)$, is the interpolated response of $\mathbf{R}(s, \vec{g}_i)$ obtained to match $\mathbf{R}(s, \vec{g}_j)$ and $\operatorname{Err}(\cdot)$ is a suitable error measure between the two responses [6]. Note that, $\alpha_{1,ij}^* = \alpha_{2,ij}^* = 1$ when $i = j$.
The evaluation of the model taken at a generic point $\vec{g}_q$ in the design space (Fig. 1) is done similarly to [6] as:

i For each *root macromodel* $\mathbf{R}(s, \vec{g}_i)$, $i = 1, \ldots, 4$, the amplitude scaling coefficient $\alpha_{1,ij}$ and frequency scaling coefficient $\alpha_{2,ij}$ are interpolated using a multilinear interpolation [21] over $\vec{g}$ at the point $\vec{g}_q$ to find $\alpha_{1,iq}$ and $\alpha_{2,iq}$. This results in the modified *root macromodels*, $\tilde{\mathbf{R}}_q(s, \vec{g}_i) = \alpha_{1,iq} \sum_{p=1}^{P_i} \frac{C_{p,i}}{s\alpha_{2,iq} - a_{p,i}} + D_i$ at $\vec{g}_q$,

ii Then the models $\tilde{\mathbf{R}}_q(s, \vec{g}_i)$, are interpolated using the multilinear interpolation [21] over $\vec{g}$ to get the final model interpolated at the points $\vec{g}_q$, $\mathbf{R}(s, \vec{g}_q)$.

This parametric macromodeling approach is performed for each region Ω_l which can either be a N-dimensional hyperrectangle or a N-simplex to cover the complete design space.

B) sequential sampling using Grid-Based Refinement

The grid-based refinement scheme of [14] works on hyperrectangular grids and generates local scalable macromodel for each and every *subspace*. The grid-based sequential sampling algorithm begins from a single subspace with 2^N corners defined by the design parameter ranges [14]. Then it finds the maximum sensitive edge by checking difference between two responses of every edge and selects the edge with maximum difference. Later, a $(N-1)$-Hyperplane perpendicular to the selected edge is used to divide the subspace into two child subspaces if the accuracy is not satisfactory. This procedure is repeated until all the subspaces are accurate and then finally a center refinement is used to complete the process as clearly explained in [14].
The idea of selecting the maximum sensitive edge is slightly modified in this work to make use of the available information generated by [6]. The idea here is to find the most

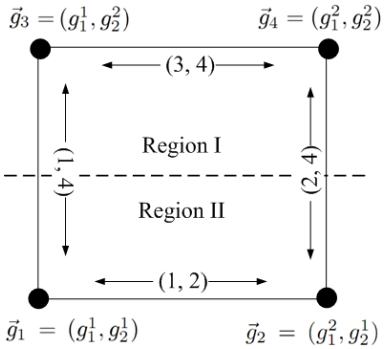

Figure 2: Subspace division along the most difficult-to-model edge.

difficult-to-model edge in terms of the macromodeling method of [6] as explained in detail below.
As in Section A), let us consider a two variable design space $\vec{g} \in (g_1, g_2)$ defined by four corners $\vec{g}_1 = (g_1^1, g_2^1)$, $\vec{g}_2 = (g_1^2, g_2^1)$, $\vec{g}_3 = (g_1^1, g_2^2)$, and $\vec{g}_4 = (g_1^2, g_2^2)$ of a rectangular region (two dimensional subspace) as in Fig. 2. In the scalable macromodeling method of [6], one scaling and another frequency shifting coefficients $(\alpha_{1,ij}^*, \alpha_{2,ij}^*)$ are calculated as in (2). The error information obtained from (2) can be used as a measure of the modeling difficulty of each and every edge $1 \leq (i,j)_{\text{edge}} \leq 4$ (see Fig. 2), and the most difficult-to-model edge is the edge with the worst-case error given by the formula,

$$(i,j)_{\text{edge}}^{\max} = \underset{(i,j)_{\text{edge}}}{\operatorname{argmax}} \left(\min_{(\alpha_{1,ij}, \alpha_{2,ij})} \left[\text{Err}(\tilde{\mathbf{R}}(s, \vec{g}_i), \mathbf{R}(s, \vec{g}_j)) \right] \right). \tag{3}$$

Then, a hyperplane perpendicular to the edge $(i,j)_{\text{edge}}^{\max}$ is used to divide the subspace into two halves. In Fig. 2, the pair $(1,4)$ was selected as the most difficult-to-model edge and a line perpendicular to that edge is used to divide the subspace into two.
However, the grid-based scheme suffers from the following:

1. The final refinement after the edge-based refinement of each subspace is performed at the center [14], and to keep the hyperrectangular nature of the grid all the lower dimensional hyperplanes such as edges, faces etc., of the subspace are divided generating a lot of points.

2. The local scalable macromodel is build by linearly interpolating 2^N *root macromodels* for every subspace [14]. Since the *root macromodel* transfer functions are appended as in [6], the order of the final scalable macromodel inside a subspace can have high values (in proportion to 2^N). This increases the evaluation time for the macromodel especially for time-domain simulations.

Both of the above mentioned issues become even more troublesome with higher dimensions. Therefore, a scattered refinement using well-conditioned simplexes becomes necessary to overcome these issues and such a scheme is presented in detail in Section III.

III Path-Simplexes

A path-simplex in $\mathbb{R}^N$, $N \in (1, 2, ...)$ is defined as an N-Simplex having N mutually orthogonal edges which, in the sense of graph theory, form a path [16]. Path-simplex has the following properties which makes it useful for the proposed sequential sampling [16]:

i. a path-simplex in $\mathbb{R}^N$ is a non-obtuse simplex and all its $(N-i)$-simplexes, $i = 1, 2, ..(N-1)$ are also path-simplexes,

ii. it contains its circumcenter ensuring good condition,

iii. path-simplex refinement is assured to be Delaunay by construction,

iv. every alternate division performed on path-simplex generates geometrically similar simplexes.

The above mentioned points ensures that slivers are never created during the local refinement of a simplex, ensuring convergence of the algorithm. Also, the Path-simplex based division has already been applied to the sequential sampling process and a preliminary work can be found in [22]. A more elaborate description is given in this paper. The proposed sequential sampling algorithm starts from an edge-based refinement scheme described above and then uses the result of [16] to refine a N-box region of the design space into $N!$ path-simplexes. Then these path-simplexes are further divided as in the Coxeter's trisection method [15] which is described below.

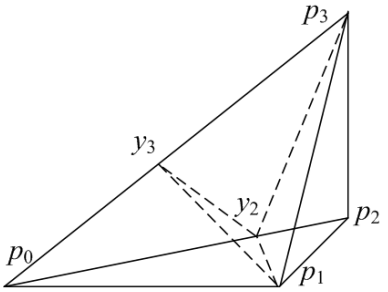

Figure 3: Coexter's trisection of the path-simplex in $\mathbb{R}^3$ (as in [15]).

In [15], Brandts et al. prove that given a path-simplex in $\mathbb{R}^N$, it can be divided into N path-subsimplexes using Coxeter's trisection method generating $N-1$ new sample points. Fig. 3 shows such a division for a path-simplex in $\mathbb{R}^3$. The corners of the path-simplex are represented by the position vectors p_0, p_1, p_2, and p_3 with respect to any arbitrary origin, and the edges $p_0 - p_1$, $p_1 - p_2$, and $p_2 - p_3$ forming a path. Three new path-simplexes are formed using the points y_2 and y_3 calculated as

$$y_j = p_j \times \frac{\|p_1\|^2}{\|p_j\|^2},\ j = 2, 3, ...N., \tag{4}$$

where, $\|.\|$ is the Euclidean norm [15]. Generation of slivers during the local refinement can be monitored by calculating the aspect ratio,

$$R_{\text{asp}} = N\frac{d}{D}. \tag{5}$$

in (5), d and D are the diameters of the inscribing and circumscribing N-spheres of the N-simplex respectively. *Root macromodels* are created at the corner points of these simplexes and using the scalable macromodeling method of [6], passive interpolated models are created for the parameterized frequency responses.
To see the advantage of using a path-simplex division, it is compared with respect to a division at the incenter of the simplex and the minimum aspect ratio (5) for each level of division is plotted in Fig. 4. The comparison is made with respect to the number of times the simplex is divided as well as the number of dimension N of the simplex. Both the division techniques start from a path-simplex of unit orthonormal edges. As seen in Fig. 4, the path-simplex division preserves the aspect ratio of the simplexes thereby assuring a good condition number. This is much better as compared to the incenter division. As stated in [15], a path-simplex if divided twice, one of the sub-sub-simplexes is similar to the original simplex keeping its aspect ratio. This can also be seen in Fig. 4, where the aspect ratio shows oscillatory behavior showing each alternate division levels are similar.

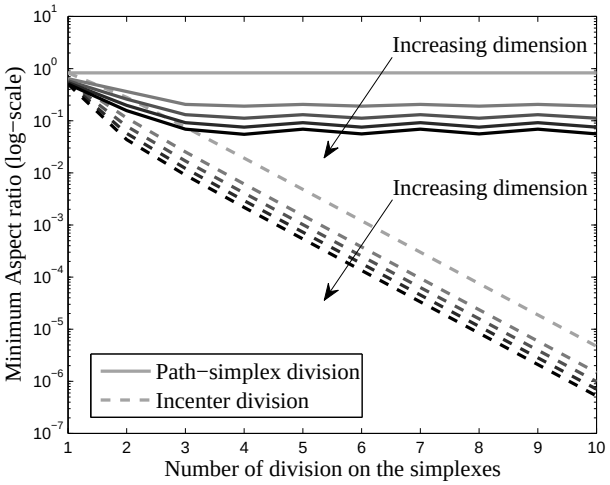

Figure 4: Comparison of Coxeter's trisection method versus incenter-based division.

Next, to show why a scattered sampling is required, some important parameters for the edge-based and center-based subspace division of [14] is compared with the path-simplex division and is tabulated in Table 1. By using a simplex-based division, the number of expensive samples generated per refinement can be brought down to $(N-1)$. Note that the number of points created per center-based division is calculated by summing all the possible hyperplanes such as edges, faces, etc., which are divided. Also, the order of the final macromodel can be reduced from that proportional to a factor of 2^N for the grid-based scheme to a factor of $(N+1)$ for the scattered scheme. This creates considerable speed-ups in macromodel evaluations both in frequency and time-domain. However, one of the issues with the path-simplex based division is that the method acts as a space-filling strategy by generating well-conditional simplexes with less emphasis on sequential sampling. Therefore a hybrid method is proposed in this paper which brings the advantages of both schemes to get a better sequential sampling strategy as will be explained in Section V.

Table 1: Comparison of different refinement strategies.

Refinement Method	# points per refinement	# regions per refinement	model order proportional to
Edge-based	2^{N-1}	2	2^N
Center-based	$1+\sum_{m=1}^{N-1}\frac{2^{N-m}N!}{m!(N-m)!}$	2^N	2^N
Path-simplex	$N-1$	N	$N+1$

IV Error estimation without validation points

In [14], the accuracy of the model is calculated by comparing the scalable macromodel with the actual EM simulations at each and every terminal subspaces. When a subspace is found to be accurate, it is not further divided and the expensive validation points from the EM solver may not used [14]. This can be tackled to a certain extend by performing an estimation of the accuracy or the error at each and every subspace as described here.

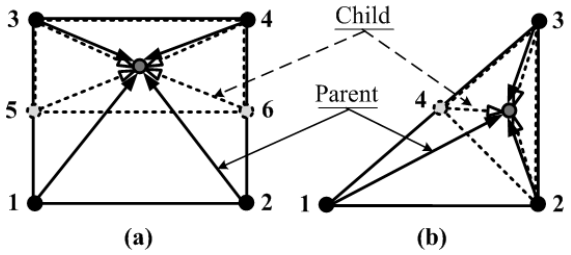

Figure 5: Error estimation with two subsequent model levels: (a) Rectangular region and (b) Simplicial region.

In this paper, a comparison between two subsequent levels of scalable macromodels is proposed to assess the convergence of the sequential sampling as shown in Fig. 5.a for a 2D case on a rectangular grid. Two different scalable macromodels are compared, one from the parent subspace (region 1-2-3-4, solid line arrows) and the other from the child subspace (region 5-6-3-4, dashed line arrows) at the center of the child subspace region. On the other hand, if the simplexes are divided, a similar strategy is used, wherein the two macromodels are compared at the incenter of the child simplex as shown in Fig. 5.b. When a convergence is observed between the two levels, the algorithm is terminated.

V Proposed sequential sampling algorithm

Fig. 6 shows the flowchart of the proposed sequential sampling algorithm. The algorithm is divided into three major categories as explained below.
Stage 1:*Initialization* is done by defining the boundaries of the design space and then generating the 2^N corner *root macromodels*. Then an initial scalable macromodel is built for this subspace by using [6] and this is the starting point of the tree-based sequential sampling. The number of EM simulations at this stage is only the 2^N corner points of the design space.

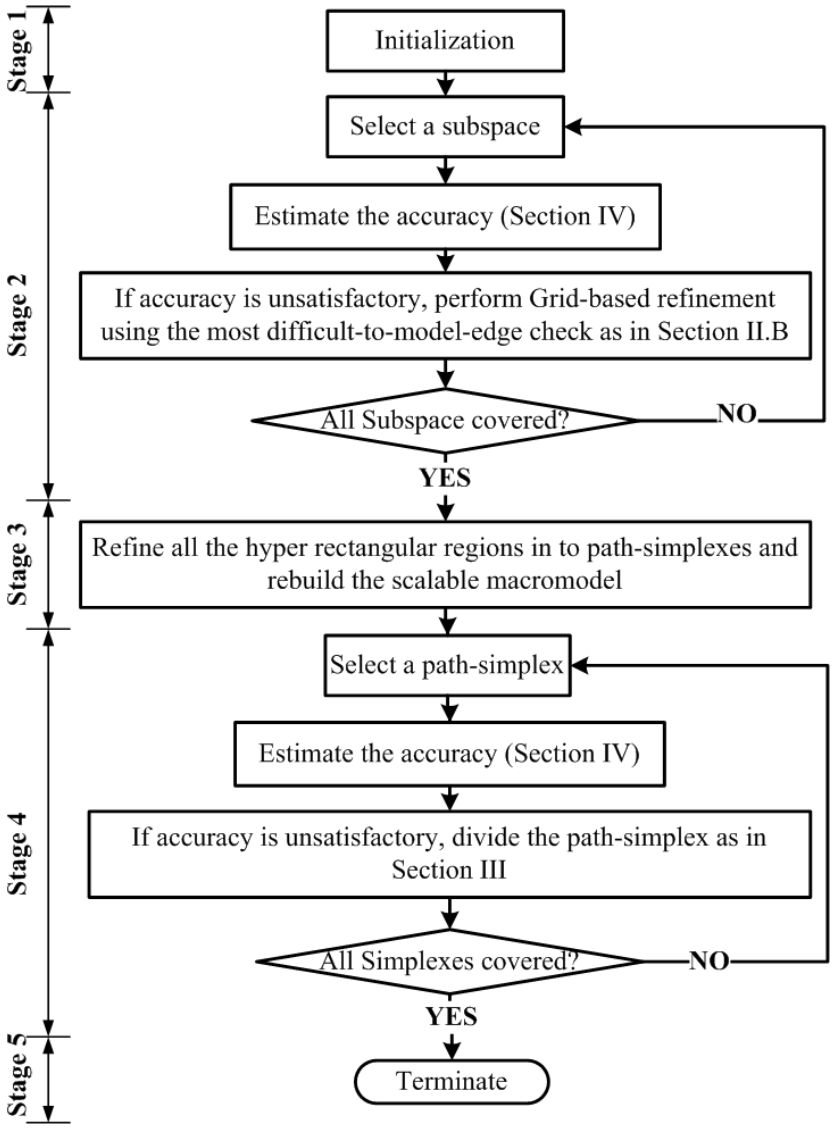

Figure 6: Flow chart of the proposed sequential sampling.

Stage 2:Then, the *grid-based refinement* is performed to refine the initial scalable macromodel using the edge refinement method as in Section B). At every iteration a subspace is selected and the modeling error is estimated using the parent-child response comparison as explained in Section IV. If the subspace is found to be inaccurate, the subspace is divided using a hyperplane perpendicular to the most difficult-to-model edge as per Section B) similar to Fig. 2. This step is continued till all the subspaces are sufficiently accurate. This initial accuracy target can only be set depending on the problem at hand. For example, considering microwave filter, if a passband requirement of -30 dB is required, the scalable macromodel should be able to describe the filter characteristics up to an accuracy of -30 dB. So, the initial modeling accuracy can be set to the bare minimum accuracy required by the designer (-30 dB in the example stated here) such that this low fidelity model can already be used the design process. The number of EM simulations at this stage depends on the grid-based refinement.

Stage 3:In the next step, the initial grid generated is rearranged into path-simplexes regions. No additional EM simulations are required at this stage. Here, the model generated using the grid-based refinement is used as the starting point and then each and every terminal subspaces are refined into path-simplexes [16]. It should be noted that, during this process only hyperrectangular regions are converted into path-simplex regions and no further calculation of scaling and frequency shifting parameters for the scalable macromodeling [6] is needed since the *root-macromodels* stays the same.
Stage 4: Finally, the *Scattered refinement* is done using the method of path-simplexes by which the final target accuracy is to be achieved. After this conversion, the simplex regions are selected and error is estimated at their incenters similar to Section IV. A higher accuracy can be selected here compared to Stage 2, but since the error is estimated between two different models (as in Setcion IV), the accuracy target at Stage 2 and Stage 4 can very well be equal and the convergence is checked by estimating the error between the two models. If the accuracy is not satisfied for some simplicial regions, they are further divided using the path-simplex refinement procedure of Section III until all the simplexes are accurate. As in the case of grid-based refinement, the number of EM simulation is decided by the scattered refinement process.
Stage 5: Once all the simplicial regions are accurate, the sequential sampling algorithm is terminated. No further EM simulations are required at this stage.
It is important to note that, when the range of the design space is increased, the algorithm takes care of the change by exploring the design space. This is done by generating additional samples (or *root macromodels*) in the newly added regions. Thus, by generating additional samples and refining the bigger regions into smaller domains, the algorithm will ensure that the scalable macromodeling scheme is able to build accurate models over the complete design space.

VI Numerical results

In this section three numerical examples are presented which demonstrate the capability of the proposed sequential sampling method for efficiently building the scalable macromodels for EM systems. For comparison purposes in terms of the computational time, all the numerical simulations have been performed on a Linux platform on Intel(R) Xeon(R) CPU E5504 @ 2.00 GHz machine with 6 GB RAM.

A) Example I: Microstrip bandpass filter

A microstrip bandpass filter on a substrate with relative permittivity $\epsilon_r = 9.0$ and a thickness of 0.660 mm is modeled in this example. The S-Parameter response of the filter is generated with the help of ADS Momentum[1]. The ADS Momentum simulation engine is used in full-wave mode. All ports are defined as single mode ports, with 50 Ω characteristic impedance. The automatic meshing (with edge mesh) uses 20 cells per wavelength, at a mesh frequency of 6 GHz. The layout of this filter is shown in Fig. 7. Two lengths L_1 and L_2 and the spacing S are chosen as design variables (see Fig. 7) in addition to frequency whose ranges are $L_1 \in [6.0, 7.0]$ mm, $L_2 \in [4.0, 5.0]$ mm,

[1]Momentum EEsof EDA, Agilent Technologies, Santa Rosa, CA.

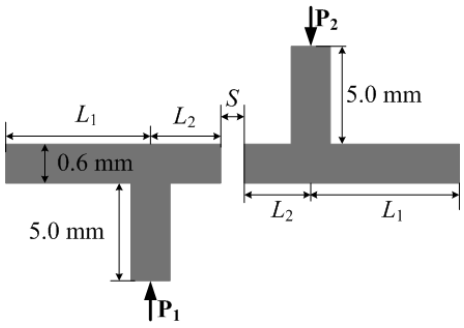

Figure 7: Example I: Layout of the microstrip bandpass filter.

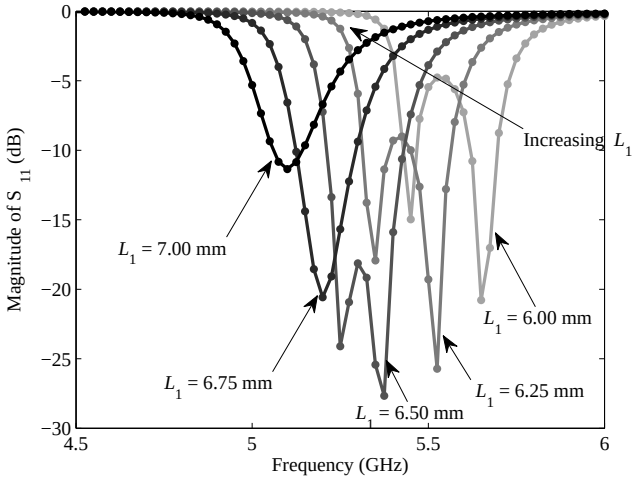

Figure 8: Example I: Magnitude of S_{11} as a function of L_1.

$S \in [0.05, 0.10]$ mm, and frequency $\in [4.0, 6.0]$ GHz. The Mean Absolute Error (MAE) measure or the $L1$-norm per port is used to assess the accuracy of the model in every N-box region of the design space:

$$E^{\mathrm{MAE}}(\vec{g}) = \max_{\substack{u=1,\ldots,P \\ v=1,\ldots,P}} \frac{1}{N_s} \left(\sum_{n=1}^{N_s} |R_{u,v}(s_n, \vec{g}) - H_{u,v}(s_n, \vec{g})| \right). \tag{6}$$

The method compares the EM simulation response $H_{u,v}(s, \vec{g})$ with the scalable macromodel response $R_{u,v}(s, \vec{g})$, where P is the number of system ports. The MAE error measure or the $L1$-norm gives a global view on the error between the two frequency responses and hence it is preferred. The target accuracy was kept at -45 dB and the initial refinement accuracy for the proposed method also kept at -45 dB.
Fig. 8 shows the parametric behavior of the magnitude of S_{11} as a function of L_1 and frequency, other values being kept at the mean value of the design space. Similarly, Fig. 9 shows the magnitude of S_{21} as a function of S and frequency. The proposed algorithm and the grid method [14] have been implemented in Matlab R2012a[2] and used to drive

[2]The Mathworks, Inc., Natick, MA, USA

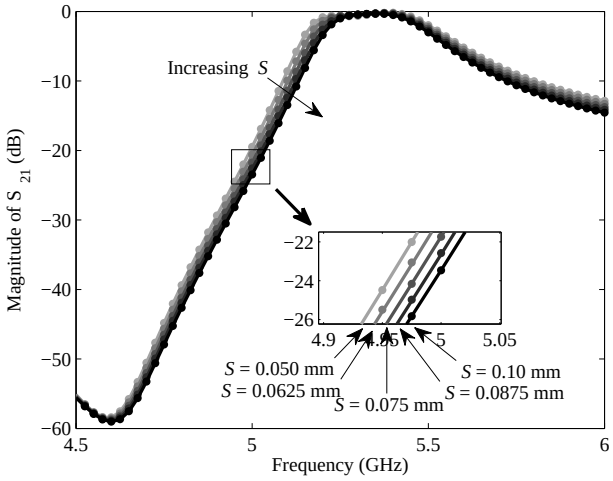

Figure 9: Example I: Magnitude of S_{21} as a function of S.

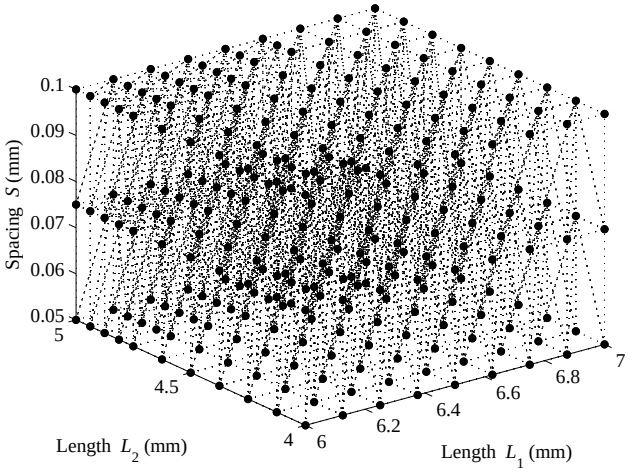

Figure 10: Example I: Sample distribution with the proposed algorithm.

the ADS Momentum simulations to generate S-responses with 31 frequency points at selected samples. The dotted curves in Figs. 8 and 9 represent the response of the scalable macromodel obtained from the proposed method. As seen a good agreement can be observed.

Fig. 10 shows the distribution of the design sample points with the proposed hybrid sequential sampling algorithm. The tessellation with the path-simplexes are also shown here. It can be observed from Fig. 10 that along the design variable L_2, the maximum number of samples are taken, which means that this variable is highly influential on the output S-parameters of the filter. The design variable S is the least influential and hence it is sparsely sampled. Also, it can be noticed that the higher values of the design variable $L_2 \in (4.5, 5.0)$ mm is more densely sampled than the other parts of the design space. This indicates a high sensitivity of the output S-parameter response of the filter to the changes

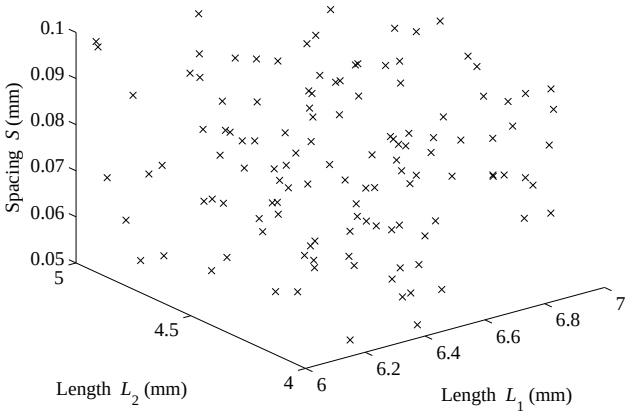

Figure 11: Example I: Verification sample distribution.

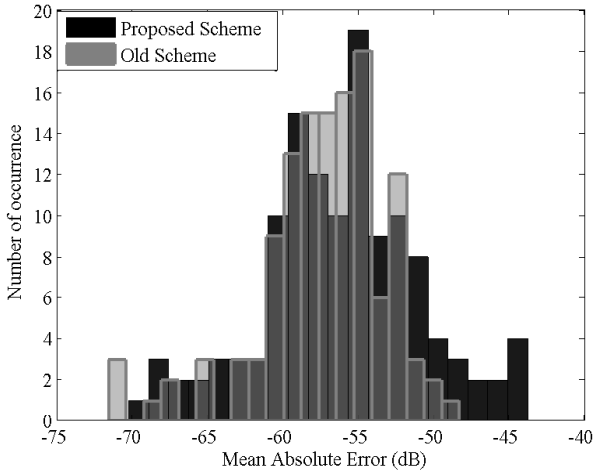

Figure 12: Example I: Mean absolute error distribution for the final verification samples.

in $L_2 \in (4.5, 5.0)$ mm in comparison with the lower values, $L_2 \in (4.0, 4.5)$ mm. In order to show the capability of the proposed algorithm, it is compared with the grid method on 125 verification points spread across the design space as shown in Fig. 11 using a Latin hypercube space filling.

Fig.12 shows the mean absolute error distribution for both sequential sampling methods over the final verification points. As seen in the figure, a comparable accuracy is achieved for the proposed sequential sampling scheme without having to use any validation points during the sampling process like the grid method. Next, to check the advantage gained by performing a hybrid algorithm, the scalable macromodeling is performed directly on the scattered sampling. That is, the Stage 2 of the algorithm in Fig.6 is not performed and the algorithm is checked for its convergence. It was observed that the algorithm, even after generating 492 samples could converge only to an accuracy level of -25.11 dB. As explained in Section III, the path-simplex division, even though very well-conditioned, is

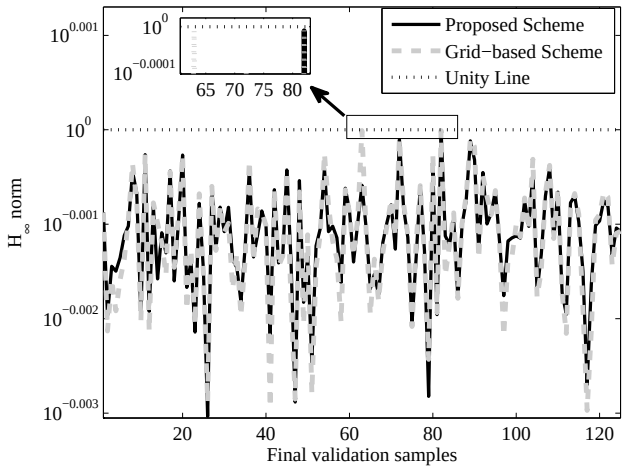

Figure 13: Example I: H_∞ norm for the final verification samples.

highly exploratory in nature and hence slow in converging to a good modeling accuracy. Thus combining the advantages of both grid-based refinement and the simplicial refinement becomes necessary to have an efficient and automated sampling strategy.
In order to check the passivity, the $\mathbf{H}_\infty$ norm $||\mathbf{R}(s,(L_1,L_2,S)||_\infty$ of the scalable macromodel was calculated for the final 125 points and is plotted in Fig.13. From the figure, it is clear that maximum norm is bounded by unity as expected. Table 2 compares the two sampling schemes over some important parameters. As seen in the table, the maximum order of the scalable macromodel is relatively high for the grid method in comparison to the proposed scheme. This is because of the fact that all the 2^N corner *root macromodels* of a subspace is augmented in the grid method whereas there are only $N+1$ corner *root macromodels* defining a simplex for the proposed scheme. This becomes more severe with higher dimensions as will be seen in a later example. The table also shows the worst case mean absolute error for the two schemes and as expected, the proposed scheme has a slightly lower accuracy level, but only requires 40% of the samples compared to grid method (please note that the target accuracy level was -45dB). This is because of the fact that the error is estimated at each level during the sampling whereas for the grid method it is validated using expensive EM simulations. This is the trade off between the two sampling schemes.
Finally, to see the speed-up gained by using a scalable macromodel in the design process, the time required for one single frequency response evaluation of the scalable macromodels are compared with the corresponding EM simulator time and the speed-up is tabulated in Table 2. The speed-up for each case is calculated by comparing the evaluation time for one frequency sweep using the macromodel (column 7 of Table 2) with the CPU time for the EM simulator. However, it should be noted that the generation of the scalable macromodel requires some initial EM simulations, but once the model is built, it can be used in multiple design optimization scenarios such as changing specifications, to make the overall design cycle very efficient.
So, in conclusion the following points can be observed from Table 2:

1. The number of samples required for the proposed scheme is reduced in comparison

Table 2: Example I: Comparison of different sampling strategies.

Sampling Method	# Samples Gen.	# Samples Val.	Maximum Order	Error (dB)	Maximum $\|\|H\|\|_\infty$	Evaluation Time (s)	Speed-up
Proposed	275	-	86	-43.8	0.999	0.136	199 ×
grid method [14]	438	248	164	-48.3	0.999	0.256	105 ×
CPU time for EM simulator to calculate a single frequency response = 27 s.							

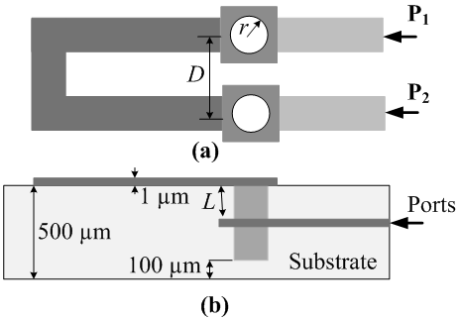

Figure 14: Example II: Layout of Microstrip with two coupled vias: (a) Top view, (b) cross sectional view.

with the methods of [14, 22].

2. The validation samples are completely removed by estimating error by comparing two different levels of the model as in Section IV.

3. Estimating the error means that the accuracy of the proposed method might not be as good as the grid method of [14]. This is the trade of between the validation points and error estimation.

4. The modeling complexity is reduced in comparison with grid method of [14] as explained before in Table 1. This also gains in the speed-up in terms of the scalable macromodel evaluation as clear from the last column of the table.

B) Example II:Microstrip with two coupled vias

A microstrip with two coupled vias on a substrate with relative permittivity $\epsilon_r = 9.0$ and a thickness of 500 μm is modeled in this example. The S-parameter response of the structure is generated with ADS Momentum. The ADS Momentum simulation engine is used in full-wave mode. All ports are defined as single mode ports, with 50 Ω characteristic impedance. The automatic meshing (with edge mesh) uses 30 cells per wavelength, at a mesh frequency of 5 GHz. Fig. 14 shows the top and cross sectional view of the structure. The length of the two vias L, the distance between the two vias D and the radius of the vias r are chosen as design variables (see Fig. 14) in addition to frequency whose ranges are $L \in [150, 250]$ μm, $D \in [1.5, 2.5]$ mm, $r \in [0.1, 0.4]$ mm, and frequency $\in [0.1, 10.0]$ GHz. The target accuracy (6) was kept at -45 dB and the initial refinement accuracy for the proposed method also kept at -45 dB. Fig. 15 shows the parametric behavior of the magnitude of S_{11} as a function of L and frequency, other

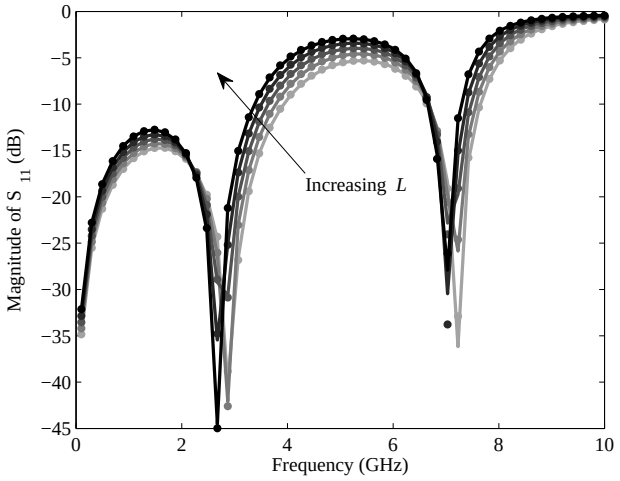

Figure 15: Example II: Magnitude of S_{11} as a function of L.

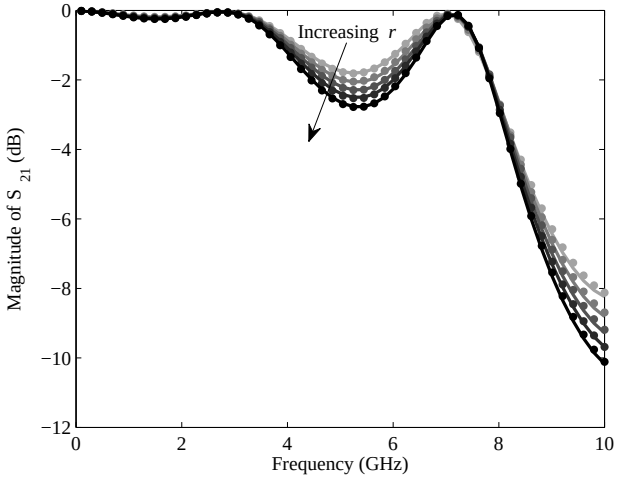

Figure 16: Example II: Magnitude of S_{21} as a function of r.

values being kept at the mean value of the design space. Similarly, Fig. 16 shows the magnitude of S_{21} as a function of r and frequency. As in Example I, the proposed algorithm and the grid method [14] have been implemented in Matlab R2012a and used to drive the ADS Momentum simulations to generate S-responses at selected samples. The dotted curves in Figs. 15 and 16 represent the response of the scalable macromodel obtained from the proposed method. As seen a good agreement can be observed.
As in the previous example, the proposed algorithm is compared with the grid method on 125 verification points spread across the design space using a Latin hypercube space filling. Fig.17 shows the mean absolute error distribution for the sequential sampling methods over the final verification points. A comparable accuracy is achieved for the proposed sequential sampling scheme as it was observed for Example I. The $\mathbf{H}_\infty$ norm $||\mathbf{R}(s,(L,D,r)||_\infty$ of the scalable macromodel was calculated for the final 125 points and

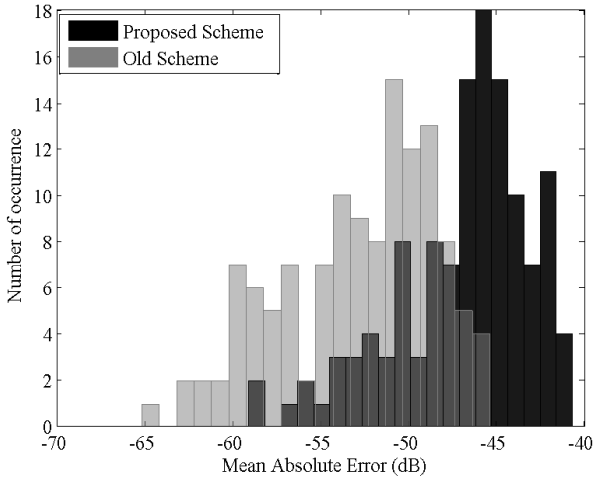

Figure 17: Example II: Mean absolute error distribution for the final verification samples.

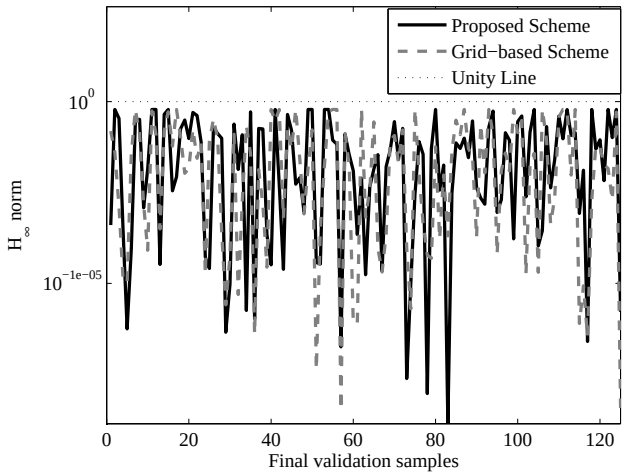

Figure 18: Example II: H_∞ norm for the final verification samples.

is plotted in Fig.18 and it is observed to be passive.
Table 3 compares the two sampling schemes over some important parameters similar to Table 2 for the first example. As for the Example I, a similar conclusion can be derived for the different quantities such as maximum model order, mean error etc. Note that the target accuracy level was -40 dB. The trade off between the two sampling schemes in terms of the error versus number of modeling samples required is also clear in the Table 3. As in the first example, to show the speed-up gained by using a scalable macromodel in the design process, the time required for one single frequency response evaluation of the scalable macromodels are compared with the corresponding EM simulator time and the speed-up is tabulated in Table 3.
So from Table 3 a similar conclusion can be derived as in the case of Example I

Table 3: Example II: Comparison of different sampling strategies.

Sampling Method	# Samples Gen.	# Samples Val.	Maximum Order	Error (dB)	Maximum $\|\|H\|\|_\infty$	Evaluation Time [s]	Speed-up
Proposed	30	-	64	-40.72	0.999	0.1180	1008 ×
Grid method [14]	92	38	128	-45.41	0.999	0.2017	590 ×
CPU time for EM simulator to calculate a single frequency response = 119 s.							

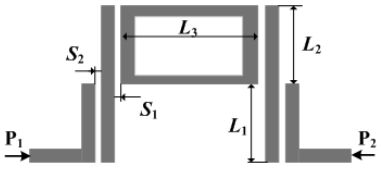

Figure 19: Example III: Layout of ring resonator bandpass filter.

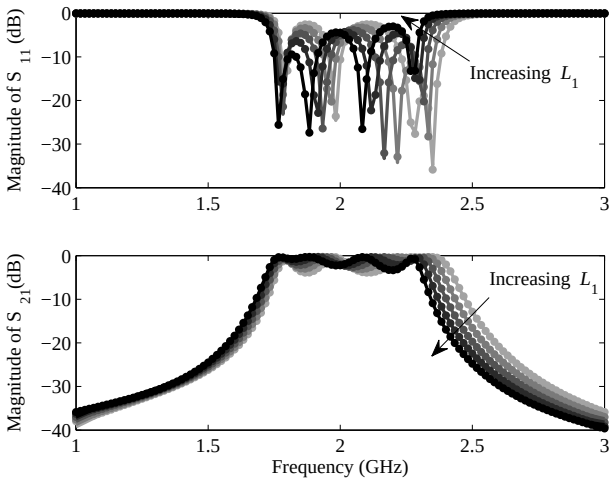

Figure 20: Example III: Magnitude of S_{11} and S_{21} as a function of L_1.

C) Example III: Ring resonator filter

A Ring resonator bandpass filter on a substrate with relative permittivity $\epsilon_r = 4.32$ and a thickness of 152 μm is modeled in this example. The layout of this filter is shown in Fig. 19. The S-parameter response of the structure is generated with ADS Momentum. The ADS Momentum simulation engine is used in full-wave mode. All ports are defined as single mode ports, with 50 Ω characteristic impedance. The automatic meshing (with edge mesh) uses 20 cells per wavelength, at a mesh frequency of 4 GHz. Two spacings S_1 and S_2 and three lengths L_1, L_2 and L_3 are chosen as design variables (see Fig. 19) in addition to frequency whose ranges are $S_1 \in [0.20, 0.30]$ mm, $S_2 \in [0.04, 0.06]$ mm, $L_1 \in [20.0, 24.0]$ mm, $L_2 \in [19.0, 21.0]$ mm, $L_3 \in [26.0, 27.0]$ mm and frequency $\in [1.0, 3.0]$ GHz. The MEA measure (6) was used to estimate the modeling accuracy. The target accuracy was kept at -40 dB and the initial refinement accuracy for the proposed method also kept at -40 dB.
Parametric behavior of some of the S-Parameter matrix entries of the filter are shown in

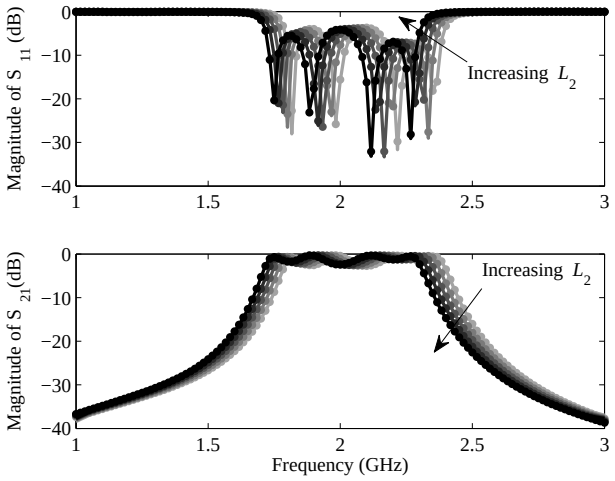

Figure 21: Example III: Magnitude of S_{11} and S_{21} as a function of L_2.

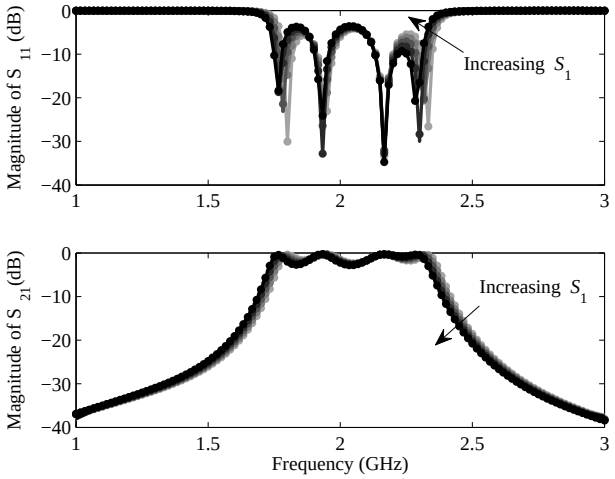

Figure 22: Example III: Magnitude of S_{11} and S_{21} as a function of S_1.

Figs. 20, 21 and 22 with respect to some design variables. In Fig. 20, the magnitude of S_{11} and the S_{21} of the filter are shown for five different values of L_1 keeping the other variables constant at their mid values of range. Similar plot can be observed in Figs. 21 and 22 for the parameters L_2 and S_1 respectively. The dotted curves in Figs. 20, 21 and 22 represent the response of the scalable macromodel obtained from the proposed method. As seen a good agreement can be observed. For this example, first a scalable macromodeling is performed using the proposed and the grid method generating two different macromodels and then as a next step, a global optimization is performed on these two macromodels generated. Finally, the modeling part as well as the optimization part for the two sequential sampling algorithms are compared in terms of important parameters.

1) Macromodeling of the ring resonator filter

As in Example I and II, The proposed algorithm and the grid method [14] have been implemented in Matlab R2012a and used to drive the ADS Momentum simulations to generate S-responses at selected samples. As in the previous examples, the proposed algorithm is compared with the grid method on 250 verification points spread across the design space using a Latin hypercube space filling. The MEA measure (6) was used to estimate the modeling accuracy.
Table 4 compares the two sequential sampling schemes over some important parameters such as the number of samples required, the worst-case accuracy over the 250 verification samples, the maximum macromodel order and the speed-up.

Table 4: Example III: Comparison of different sampling strategies.

Sampling	# Samples		Maximum	Error	Evaluation	Speed-up
Method	Gen.	Val.	Order	(dB)	Time [s]	
Proposed	306	-	156	-42.92	0.3156	434 ×
Grid method [14]	405	66	832	-42.03	1.9584	70 ×
CPU time for EM simulator to calculate a single frequency response = 137 s.						

So, as in the case of Example I and II, the following can be observed from the table:

i. considerable reduction in the number of samples and consequently in overall complexity for the proposed scheme over the grid method,

ii. computational complexity of the scalable macromodel generated using the grid method is higher than the proposed scheme as explained in Section III in Table 1, and finally

iii. Comparable accuracy for the proposed method and the grid method is observed.

2) Optimization of the ring resonator filter

The two scalable macromodels generated using the grid method and the proposed method were used in a design optimization scenario. The design specifications of the filter are given in terms of the scattering parameters S_{21}:

$$|S_{21}| > -2.0\,\text{dB} \quad \text{for } 1.75\,\text{GHz} \leq freq \leq 2.25\,\text{GHz} \tag{7a}$$

$$|S_{21}| < -25\,\text{dB} \quad \text{for } freq < 1.5\,\text{GHz}\,,\ freq > 2.5\,\text{GHz} \tag{7b}$$

From the design specifications (7), a cost function is formulated in terms of S_{21} and frequency. A global optimization method based on the DIviding RECTangle (DIRECT) strategy [23] is used to minimize the cost function. The method [23] balances between a global and local search and finds an optimization solution $\vec{g}^* = (L_1^*, L_2^*, L_3^*, S_1^*, S_2^*)$. Since the cost functions are generated with the help of the scalable macromodels generated with the grid method [14] and the proposed sampling scheme, the evaluation time is much less in comparison with the actual ADS simulations.
It can be observed from Table 5 that the average macromodel evaluation time per sample for the grid method is relatively higher in comparison with the proposed scheme. This is

Table 5: Example III: Optimization results.

Sampling method	Proposed	Grid method [14]
# model evaluations	802	802
Optimization time (s)	311.6	2099.5
Final solution $\vec{g}^*$ (mm)	[22.96, 21.00, 26.67, 0.30, 0.06]	[22.96, 21.00, 26.67, 0.30, 0.06]
Optimum cost	-0.0018	-0.0018

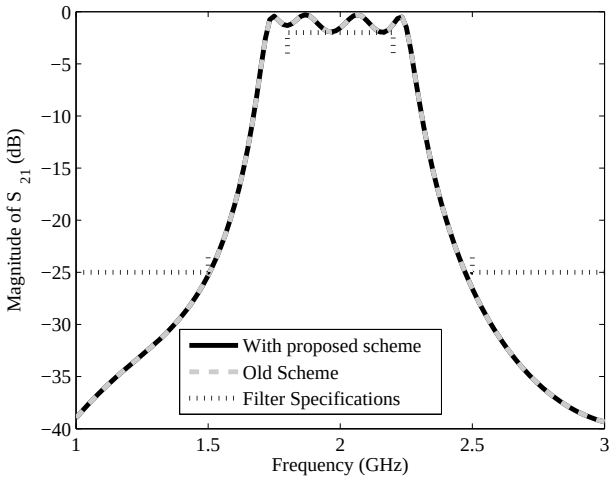

Figure 23: Example III: ADS Momentum response at the optimal solutions generated with the two macromodels.

indeed because of the increasing complexity of the macromodel generated using the grid-based interpolation in comparison with the scattered interpolation as described before (Table. 4). However, both the macromodels finds a single optimum and is also verified with ADS Momentum simulation and is shown in Fig. 23. In the figure the requirements (7) are shown by the dotted lines. The two responses generated by the ADS Momentum at the solutions given in Table 5 are also shown in Fig. 23 which satisfy the requirements.

VII Conclusion

A hybrid sequential sampling scheme on scattered grids for automatic construction of scalable macromodels for microwave systems is presented in this paper. The method uses a passivity-preserving scalable macromodeling method along with a sampling based on well-conditioned simplicial refinement strategy to build the scalable macromodels. The method is also compared with previous hyperrectangular sampling method in terms of several important parameters. Three pertinent numerical examples show the modeling capability of the proposed sequential sampling method with less computational resource requirements. One of the example also demonstrates a design optimization scenario using the generated macromodels. These numerical examples show the advantages of using the proposed method in generating accurate scalable macromodels automatically from the EM simulation

data, over the design space of interest, with minimum computational resources.

ACKNOWLEDGEMENTS

This work was supported by the Research Foundation Flanders (FWO) and by the Interuniversity Attraction Poles Programme BESTCOM initiated by the Belgian Science Policy Office.

REFERENCES

[1] F. Ferranti, L. Knockaert, and T. Dhaene, "Parameterized S-parameter based macromodeling with guaranteed passivity," *IEEE Microwave and Wireless Component Letters*, vol. 19, no. 10, pp. 608–610, Oct. 2009.

[2] ——, "Guaranteed passive parameterized admittance-based macromodeling," *IEEE Transactions on Advanced Packaging*, vol. 33, no. 3, pp. 623–629, Aug. 2010.

[3] P. Triverio, M. Nakhla, and S. Grivet-Talocia, "Passive parametric macromodeling from sampled frequency data," *IEEE International Conference on Signal Propagation and Interconnects*, pp. 117 –120, May 2010.

[4] F. Ferranti, L. Knockaert, T. Dhaene, and G. Antonini, "Passivity-preserving parametric macromodeling for highly dynamic tabulated data based on Lur'e equations," *IEEE Transactions on Microwave Theory and Techniques*, vol. 58, no. 12, pp. 3688 –3696, Dec. 2010.

[5] F. Ferranti, L. Knockaert, and T. Dhaene, "Passivity-preserving interpolation-based parameterized macromodeling of scattered s-data," *IEEE Microwave and Wireless Components Letters,*, vol. 20, no. 3, pp. 133 –135, March 2010.

[6] ——, "Passivity-preserving parametric macromodeling by means of scaled and shifted state-space systems," *IEEE Transactions on Microwave Theory and Techniques*, vol. 59, no. 10, pp. 2394 –2403, Oct. 2011.

[7] S. Peik, R. Mansour, and Y. Chow, "Multidimensional Cauchy method and adaptive sampling for an accurate microwave circuit modeling," *IEEE Transactions on Microwave Theory and Techniques,*, vol. 46, no. 12, pp. 2364 –2371, Dec. 1998.

[8] A. Lamecki, P. Kozakowski, and M. Mrozowski, "Efficient implementation of the Cauchy method for automated CAD-model construction," *IEEE Microwave and Wireless Components Letters,*, vol. 13, no. 7, pp. 268 –270, July 2003.

[9] A. Cuyt, R. Lenin, S. Becuwe, and B. Verdonk, "Adaptive multivariate rational data fitting with applications in electromagnetics," *IEEE Transactions on Microwave Theory and Techniques,*, vol. 54, no. 5, pp. 2265 – 2274, May 2006.

[10] V. Devabhaktuni, B. Chattaraj, M. Yagoub, and Q.-J. Zhang, "Advanced microwave modeling framework exploiting automatic model generation, knowledge neural networks, and space mapping," *IEEE Transactions on Microwave Theory and Techniques,*, vol. 51, no. 7, pp. 1822 – 1833, July 2003.

[11] A. Lamecki, L. Balewski, and M. Mrozowski, "Adaptive CAD-model construction schemes," *IEEE Transactions on Magnetics,*, vol. 45, no. 3, pp. 1538 –1541, March 2009.

[12] P. Basl, R. Gohary, M. Bakr, and R. Mansour, "Modelling of electromagnetic responses using a robust multi-dimensional Cauchy interpolation technique," *IET Microwaves, Antennas Propagation,*, vol. 4, no. 11, pp. 1955 –1964, Nov. 2010.

[13] R. Rohrer and H. Nosrati, "Passivity considerations in stability studies of numerical integration algorithms," *IEEE Transactions on Circuits and Systems,*, vol. 28, no. 9, pp. 857 – 866, Sep. 1981.

[14] K. Chemmangat, F. Ferranti, T. Dhaene, and L. Knockaert, "Scalable models of microwave system responses using sequential sampling on unstructured grids," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 2013.

[15] J. Brandts, S. Korotov, and M. Krizek, "Dissection of the path-simplex in $\mathbb{R}^n$ into n path-subsimplices," *Linear Algebra and its Applications*, vol. 421, no. 23, pp. 382 – 393, 2007.

[16] J. Brandts, S. Korotov, M. Krizek, and J. Solc, "On nonobtuse simplicial partitions," *SIAM Review*, vol. 51, no. 2, pp. 317–335, 2009.

[17] X. Li, "Generating well-shaped d-dimensional Delaunay meshes," *Computing and Combinatorics*, pp. 91–100, 2001.

[18] A. Constantiniu, P. Steinmann, T. Bobach, G. Farin, and G. Umlauf, "The adaptive delaunay tessellation: a neighborhood covering meshing technique," *Computational Mechanics*, vol. 42, no. 5, pp. 655–669, 2008.

[19] B. Gustavsen and A. Semlyen, "Rational approximation of frequency domain responses by vector fitting," *IEEE Transactions on Power Delivery*, vol. 14, no. 3, pp. 1052–1061, July 1999.

[20] B. Gustavsen, "Fast passivity enforcement for S-parameter models by perturbation of residue matrix eigenvalues," *IEEE Transactions on Advanced Packaging*, vol. 33, no. 1, pp. 257–265, Feb. 2010.

[21] A. Weiser and S. E. Zarantonello, "A note on piecewise linear and multilinear table interpolation in many dimensions," *Mathematics of Computation*, vol. 50, no. 181, pp. 189–196, Jan. 1988.

[22] K. Chemmangat, T. Dhaene, and L. Knockaert, "Scalable macromodelling of microwave system responses using sequential sampling with path-simplexes," *Electronics Letters*, vol. 49, no. 15, pp. 950–952, 2013.

[23] Y. D. Sergeyev and D. E. Kvasov, "Global search based on efficient diagonal partitions and a set of lipschitz constants," *SIAM Journal on Optimization*, vol. 16, no. 3, pp. 910–937, 2006.

Krishnan Chemmangat was born in Kerala, India, on December 28, 1984. Krishnan received the Bachelor of Technology (B.Tech) degree in Electrical and Electronics engineering from the University of Calicut, Kerala, India in 2006 and the Master of Technology

(M.Tech) degree in Control and Automation from the Indian Institute of Technology Delhi, New Delhi, India in 2008. Since April 2010 he is active as a PhD student in the research group Internet Based Communication Networks and Services (IBCN) at Ghent University. His research interests include parametric macromodeling, sequential sampling and sensitivity analysis.

Tom Dhaene was born in Deinze, Belgium, on June 25, 1966. He received the Ph.D. degree in electrotechnical engineering from the University of Ghent, Ghent, Belgium, in 1993. From 1989 to 1993, he was Research Assistant at the University of Ghent, in the Department of Information Technology, where his research focused on different aspects of full-wave electro-magnetic circuit modeling, transient simulation, and time-domain characterization of high-frequency and high-speed interconnections. In 1993, he joined the EDA company Alphabit (now part of Agilent). He was one of the key developers of the planar EM simulator ADS Momentum. Since September 2000, he has been a Professor in the Department of Mathematics and Computer Science at the University of Antwerp, Antwerp, Belgium. Since October 2007, he is a Full Professor in the Department of Information Technology (INTEC) at Ghent University, Ghent, Belgium. As author or co-author, he has contributed to more than 300 peer-reviewed papers and abstracts in international conference proceedings, journals and books about computational science and engineering, numerical analysis, and computer science. He is the holder of 5 U.S. patents.

Luc Knockaert received the M. Sc. Degree in physical engineering, the M. Sc. Degree in telecommunications engineering and the Ph. D. Degree in electrical engineering from Ghent University, Belgium, in 1974, 1977 and 1987, respectively. From 1979 to 1984 and from 1988 to 1995 he was working in North-South cooperation and development projects at the Universities of the Democratic Republic of the Congo and Burundi. He is presently affiliated with the Interdisciplinary Institute for BroadBand Technologies, iMinds (www.iMinds.be) and a professor at the Dept. of Information Technology, Ghent University (www.intec.ugent.be). His current interests are the application of linear algebra and adaptive methods in signal estimation, model order reduction and computational electromagnetics. As author or co-author he has contributed to more than 120 international journal and conference publications. He is a member of MAA, SIAM and a senior member of IEEE.